Federal Student Aid (FSA)
Financial Management System (FMS)

FFEL Lender Reporting System Redesign

Test Approach: LaRS

Creation Date:	4/10/02
Last Updated:	5/28/02

Table of Contents

Document File Name

This document can be found at the following location on the Accenture X Drive:
X:\FINANCIAL PARTNERS\TO 73 - Lender Redesign\Testing

Document Revision History

April 18, 2002	Final document delivered to Nancy Krecklow, Angela Baker, Frank Ramos, Todd Elliot and Jen Alden

Distribution

Copy No.	Name	Location

Executive Summary

This document describes how the LaRS and Accounting subsystems will be tested prior to being migrated into a production environment. Topics covered include unit testing by programmers, functional testing, performance testing, and acceptance testing.

1. Unit testing will be performed by the programmers in the development environment
2. A functional testing team consisting of LaRS and FMS representatives will review the requirements document and identify existing scripts that can be used as is or with some modifications to test the requirements
3. LaRS and FMS team will work together to develop scripts and test common functions
4. The functional testing team will perform a gap analysis between the requirements and the existing scripts and identify new scenarios that will need to be tested
5. The functional testing team will write new test scripts and modify existing scripts
6. The functional testing team will identify data requirements (specific reference data that must be established prior to testing)
7. A test environment (instance) will be established, and reference data loaded into it
8. The development team lead will work with the testing team lead to migrate data into the test instance
9. The functional testing team will manually execute the pre-written test scripts
10. The testing team and development team will work to resolve issues
11. The software will be functionally accepted, and migrated using configuration management standards

1 INTRODUCTION

1.1 PURPOSE & SCOPE

This document describes the overall testing strategy for the Lender Payment Process Redesign project. The new, automated quarterly reporting process requires the development of the Lender Reporting System (LaRS) application for the Federal Student Aid's (FSA's) Financial Management System (FMS). This application will undergo unit, system, integration, and user acceptance testing, before being made available to FSA FMS users and the FFEL community. Successful completion of these testing phases will ensure that both new applications meet both the business needs of the end users as well as the functional requirements specified in the Application Design Document. The objectives of the test are to:

- Ensure that a quality product is delivered to the Community
- Minimize risk
- Find and fix problems early in the process
- Follow Capability Maturity Model (CMM) guidelines and FMS defined procedures

1.2 BACKGROUND

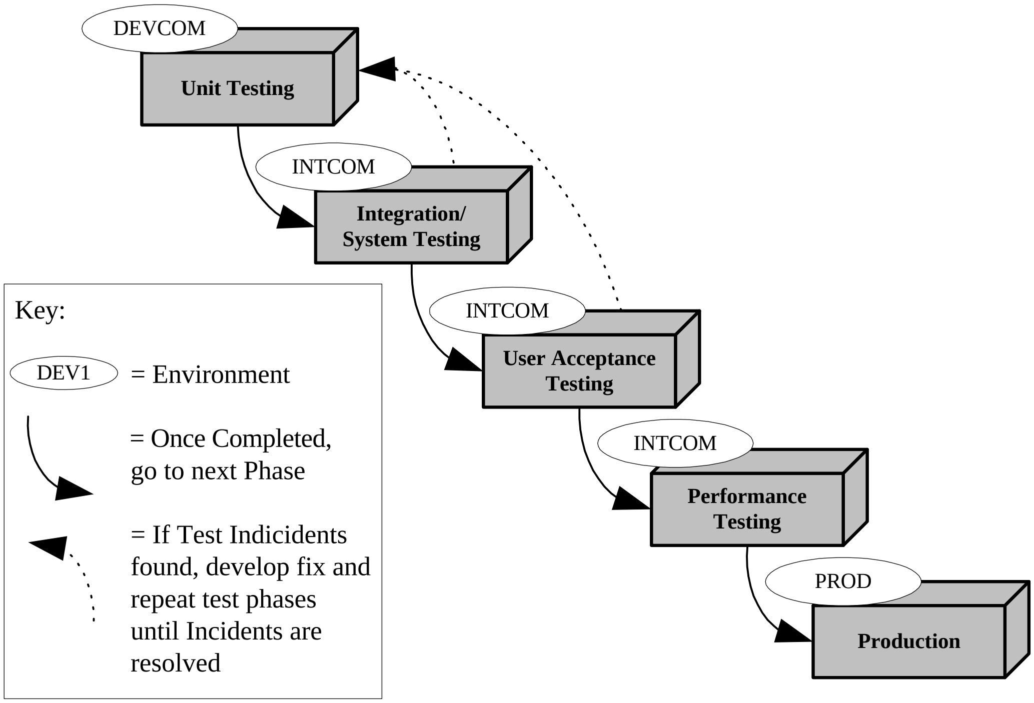
LaRS is being developed as part of Financial Partners (FP) redesign efforts. The goal of the LaRS program is to develop a system that will streamline the business processes associated with exchanging financial information with Lenders and Servicers. This involves developing new business activities that integrate Lender and Servicer reporting with FMS. These business activities include: allowing Lenders and Servicers to electronically complete and submit their reports, developing digital approval and ad hoc query functionality, complying with current legislative mandates, providing timely notification of errors following submission, and integrating all activities with concurrent FSA enterprise-wide initiatives.

Before Lenders and Servicers can use LaRS to submit their quarterly reports, they must first apply to the program using the Web-enabled LAP system. Such a system increases program integrity and efficiencies while decreasing errors related to manual processes used previously. LAP provides an automated method for FSA FP to review and accept user data. Additionally, LAP is used as a way for current Lenders and Servicers to perform an online validation of their demographic data in preparation for their transition into the LaRS program.

2 TEST OVERVIEW

This section defines the types of testing that will take place (unit, functional/integration, and acceptance testing), and the effort to conduct functional/integration testing.

Several phases of testing will be conducted to ensure that the automated reporting process meets the business needs of FSA. The testing will also ensure that the new applications meet the requirements specified in the Application Design Document (see Appendix 3.5 "Requirements Matrix"). In all phases of testing, the Lender Redesign Test Team will be responsible for identifying test incidents and communicating them to the Development Team. The figure below shows the relationships between the various phases of testing that will be conducted. These phases are discussed in more detail in Section.


DEVCOM
Unit Testing
INTCOM
Integration/ System Testing
INTCOM
User Acceptance Testing
INTCOM
Performance Testing
PROD
Production
Key:
DEV1
= Environment
= Once Completed, go to next Phase
= If Test Indicidents found, develop fix and repeat test phases until Incidents are resolved

2.1 Unit Testing

The Lender Redesign Development Team will conduct unit testing to ensure that each developed code module meets its particular business needs and requirements. Unit testing will be conducted in the DEVCOM environment.

2.2 Integration / System Testing

The Lender Redesign Test Team will conduct integration testing on LaRS to ensure that all developed code modules work together to meet the intended business needs and requirements. Integration testing will also ensure that the LaRS application works with the changes developed as part of the FSA FMS Phase IV release.

LaRS will conduct system tests to ensure that the components of each system work together as indented. Both integration and system testing will be conducted in the DEVCOM environment.

2.2.1 Number of Levels of Integration Testing

Good testing practices indicate that the following levels of testing should be undertaken during integration testing:

- Environment setup—verifying code migration, string testing to ensure that basic functionality works, data setup, *etc.* – 5% of testing effort
- Basic functionality / normal processes – 75% of testing effort
- Exception cases – 20% of testing effort

2.2.2 Number of Passes for Each Level

1 First pass—manual

2 Second pass (if needed) —manual, to re-test failed items from first pass

3 Third pass (if needed) – manual, to re-test failed items from the second pass. Anything more than 3 passes probably indicates a serious quality problem, and will be reported to the Project Team Lead.

2.3 User Acceptance Testing

The Lender Reporting Test Team, FSA client personnel, and the Lender / Servicer community will conduct user acceptance testing to ensure that the LAP and LaRS applications are ready for implementation. User acceptance indicates that the applications are ready for deployment to the production-level system, and is the final step in the process of development and testing. User acceptance testing will be conducted in the INT1 environment.

2.4 Regression Testing

Automated regression testing by the ITA team will take place *after* the software has completed System Test and is considered stable. Given the combination of (1) large number of users, (2) data volume (3) new business process that will result from this implementation, it would be wasteful to perform automated testing until the software has been fully system tested including interface verification.

2.5 Performance Testing

Performance test is a mechanism for determining an application's performance behavior under load (usage). A goal is usually set for each test, ie number of users or transactions. Most often, the number of users or transactions is defined based on the maximum usage requirement. Performance Test also allows for finding bottleneck in an application; it is another way to fine tune an application. For applications in development, Performance Test can be used for production capacity planning as well.

3 Level Of Effort

3.1 Determine Total Testing Effort

Using metrics gathered during previous FMS testing efforts, an estimate of the LaRS testing effort in man-hours has been developed.

1. Review Detailed Designs; determine which conditions can by tested by (1) existing scripts "as is," (2) existing scripts modified, and (3) new scripts
2. Estimate completion time for scripts based on complexity (High, Medium, or Low)
3. Factor in the effort to re-test of failed scripts, capture general ledger postings, *etc.*

As of 4/18/2002, we estimate that testing will be a 4 to 3 person-month effort, depending upon the expertise of the test team. Supporting details for this effort can be found in Section 9.2.

3.2 Determine Available Resources

1. Determine desired composition of team and skill level of resources (need script writers, script executors, task manager)
   - Functional experts (LaRS and FMS representatives)
   - Testing experts
2. Gather estimate from FMS team on available hours
3. Receive commitment from FMS for functional test team
4. Allow for external factors (other time commitments)
5. Prepare proposal on staffing hours for both government and MOD Partner

3.3 Determine Test Duration

Based on a 4-person team, testing (writing and executing scripts) should take between 3 and 3.2 months. A 4-person team is the practical maximum team size; anything larger would likely result in conflicts.

4 Develop Scripts

The process for developing integration test scripts will be as follows:

1. Identify Test Scenarios from Requirements documents
2. Identify Test Conditions from Detailed Design documents
3. Map Test Conditions to Requirements
4. Write new scripts or modify existing scripts

4.1 Identify Test Scenarios

4.1.1 Review Requirements Documents

Identify requirements, from the High-Level Requirements document (for maintenance screens), or the Detailed Functional and Technical Requirements documents (for all other screens). Specify the source of the requirement, if known (Profile Tech Design, *etc.*). For example, Allow User to Select Invoice Year,

> "The system will allow the user to select invoice year from a list of values. The system will provide the ability to prevent a user from selecting an invalid year."

The functional test must verify that the software supports this requirement.

4.1.2 Review Existing Scripts

Existing Account Receivable and Payable scripts should be reviewed to determine how many of the new requirements are covered by existing scenarios.

4.1.3 Perform Gap Analysis

Gaps between the requirements that need to be tested and the existing scripts need to be identified. New scenarios must be identified to fill in the gaps.

4.2 Identify Test Conditions

Using the detailed design documents, identify business rules and field edits, then drill down one level deeper to identify test conditions that will validate those rules and edits. For example, a screen might have a rule that states that if Field 1 contains value "A," then the pick list for Field 2 consists of values X, Y and Z, and Field 3 must be left blank. If Field 1 contains value "B," then the pick list for Field 2 consists of values J and K, and Field 3 must contain a numeric value. The following conditions could be derived from these rules and edits:

Field 1 value	Field 2 value	Field 3 value	Expected Result
A	X	Blank	Pass
A	Y	Blank	Pass
A	Z	Blank	Pass
A	X	9	Fail
A	Y	9	Fail
A	Z	9	Fail
B	J	9	Pass
B	K	9	Pass
B	J	Blank	Fail
B	K	Blank	Fail

Table 4-1: Sample Test Conditions

Obviously, there is a fair amount of redundancy in these conditions, and it is up to the test analyst to determine the best way to streamline the scripting process. For example, if Field 2 is informational only and the value entered does not affect system processing, then it is probably not necessary to test every value; the user could test value "X" and not test values "Y" and "Z" and the test would be just as thorough.

These test conditions should be entered into the Microsoft Access tracking database.

4.3 Map Test Conditions to Requirements

The test conditions should be mapped to the requirements.

4.4 Write Scripts

The test team leader will work with the test team to create new scripts and modify existing scripts. Reports can then be generated and given to the test team showing the conditions that a script should test, and the tester will write or modify a script to cover those conditions. Scripts should be consolidated, where possible, to maximize test scenarios and minimize staff hours.

4.4.1 Test Data

Sample test data will be created based on previously submitted quarterly reports from FFEL Community Lenders and Servicers. Test data will be used to test normal processing, error processing and reasonability checking. Demographic Data from Lenders and Servicers will also be used when testing login functionality. All Possible Lender / Servicer combinations will be conducted as part of the testing effort. These combinations include:

	LID(s)		**Servicer**			**Trustee**	**Submission Format**		
	One LID	Multiple LIDs	No Servicer	One Servicer	Multiple Servicer	Trustee	Web	Paper	Electronic
Lender Type 1	x		x				x		
Lender Type 2	x		x					x	
Lender Type 3	x		x						x
Lender Type 4	x			x			x		
Lender Type 5	x				x		x		

	LID(s)		Servicer			Trustee	Submission Format		
	One LID	Multiple LIDs	No Servicer	One Servicer	Multiple Servicer	Trustee	Web	Paper	Electronic
Lender Type 6		x	x				x		
Lender Type 7		x		x			x		
Lender Type 8		x			x		x		
Lender Type 9		x			x				x
Lender Type 10	x					x	x		
Lender Type 11		x				x	x		

4.4.2 Modify Existing Scripts

Existing scripts should be modified to take into account the differences between the Raytheon system, and the LaRS system. In addition, existing scripts should be modified to cover requirements that did not exist prior to this redesign effort, and thus were not tested by the old scripts.

4.4.3 Write New Scripts

New scripts should be written to cover the scenarios identified in the gap analysis.

4.5 Numbering Schemes

1. Each Requirement is listed by the unique section number in the Detailed Functional Requirements Document.
2. Each Business Rule and Field Edit is given a unique sequential number (1…*n*) .
3. Each Script is numbered as follows: “LR” + function tested + sequential number. For example, LRMaint01 is the first script that tests LR Maintenance screens.

4.6 Overview of Script Development Database Tool

The project team has developed a tool in Microsoft Access to help develop test scripts and ensure that the LaRS application is thoroughly tested. Using this tool, it will be possible to:

1. List business rules and field edits, as defined in the Functional Detailed Design and Technical Detailed Design
2. Track whether a script has been tested or not
3. Record the dates on which a scripts passed, failed, or was re-tested
4. Group scripts by cycle
5. Identify the type of screen or function that a script tests: maintenance, transactions, programs, looks-ups, or reports

This database can be modified, and new queries and reports developed, as needed during the testing effort.

4.7 Develop Script Standards

Please see 9.3 for an example of what the LaRS functional scripts will look like. Each script contains the following information:

1. Script Number – assigned by script writer
2. Description
3. Created By
4. Created Date
5. Tested By – written on the printed script by the tester
6. Test Date – written on the printed script by the tester
7. Prerequisites – reference data or transactions that must exist prior to script execution
8. Conditions Covered

For each step within the script, the following information is shown in table format:

9. Step number – auto-assigned by the script template
10. Cond. Num. – references the unique condition number from the Microsoft Access database
11. Test Condition – summary of what the step does
12. Action – detailed instructions for how to accomplish the step
13. Expected Results – how the system should respond to the action
14. Pass/Fail – written on the printed script by the tester
15. Actual Results
16. Comments
17. SIR #
18. Requirement Reference

4.8 Test Readiness Review

Define Requirements Test conditions, scripts, scripts execution timeframes and testing phases will be reviewed during the Test Readiness Review. All parties involved or impacted by the LaRS testing effort will attend the test readiness review. Attendees include FSA team leads, the LaRS team, FMS Phase IV testing team and IV&V.

4.9 Define Requirements

Reference data must be defined and loaded into the system prior to testing. This data includes:

1. LAP Data – User profile detail
2. Interest Rate Data – Tables 17, 18 & 30
3. User Setup – user roles and restrictions

LAP Data can loaded manually using the LAP conversion script in the test environment.

4.10 User Setup

We must establish the following User ID Codes for each tester:

- LAN ID, including default directory access, printer locations
- Oracle ID

5 Execute Test Scripts

5.1 Identify and Reserve Testing Facility

Testing will be conducted by the test team in a pre determined location. Experience has shown that having all testers in the same room leads to quicker issue resolution and greater learning than having the testers dispersed. Start date TBD, pending final approval of test scripts.

5.2 Conduct Familiarization Training

The test team leader will brief the testers on all aspects of the functional testing process, including logging on to the system, menu navigation, standard reference values, how scripts will be assigned, how to execute a script, and issue resolution.

5.3 Distribute the Scripts

The test team leader will assign scripts to the test team to optimize throughput, taking into consideration such factors as team availability, database issues, software delivery schedules, date dependencies, *etc.*

5.4 Execute Scripts

The test team, under the supervision of the test team leader, will execute the scripts according to the step-by-step directions on each script. Scripts that pass will be signed off and stored in a binder. Scripts that fail should be dealt with as described in the following section.

5.5 Identify and Resolve Issues

The following process will be followed for identifying and resolving problems encountered during testing.

1. Tester discovers a discrepancy with a script and works with another tester to determine if the problem is with the software, the test script, or some other source. This will serve as a form of triage to identify true issues, and to not bother the programmers with issues not related to the code.

2. If the test team determines that the problem is with the software, the tester log the issue in the Test Database with supporting detail including, screen prints or sections of the requirements document. The tester documents the problem by writing “Fail” in the Pass/Fail column of the script, at the step where the script failed. The tester should also write the date and time the failure was noted.

3. The Microsoft Access database should be updated to reflect the script failure. In addition to tracking software failures, the team will also track failures related to undefined requirements, faulty scripts, Oracle errors, and “other.”

4. The development team lead should assign the problem to a developer to correct it.

5. The developer should work with the tester in the development environment to ensure that the software change fixed the problem.
6. The development team lead should notify the testing team lead that the changed software will be migrated to the test environment. This should be done following the FMS process.
7. The tester will confirm, in the test environment, that the software change fixed the problem.
8. The tester will note, in the Pass/Fail column of the script, the date and time that the step of the script was successfully executed, and continue with the script.

During this process, if a problem is identified in testing, the test and development team leads will make a determination on the need for and scope of regression testing.

The key to quickly resolving software defects is direct communication between the tester and the developer who is fixing the problem. The development team lead and the test team lead should be kept in the loop, but must not become a bottleneck.

5.6 Monitor and Report Upon Testing Progress

Testing progress will be tracked by business function, rather than module, because certain functions cannot be tested until additional modules are delivered to the testing team.
For example, For LaRS: The report function can not be tested prior to the invoice creation process. For AR/AP: Receivables screen, which is used to establish a receivable, will be delivered to the testing team prior to the Collection screen. Much of the functionality related to establishing a receivable can be tested prior to testing the Collection screen; however, certain edits are not invoked until a receivable has been fully collected, and it won't be possible to test these until the Collections screen is delivered. If we track our progress based on business function, then we can state that every function of the Receivables screen has been tested *except* for those related to fully-collected receivables. If, on the other hand, we track our progress by module, then the Receivables screen as a whole cannot be considered fully tested until all modules that affect it have been delivered and tested.

The Microsoft Access script development database will be used as a tracking tool to keep current on:

1. Total scripts to test
2. Scripts tested to date
3. Tester name
4. Date tested
5. Pass/Fail
6. Problems found, by category (software, database, ill-defined requirements, *etc.*)
7. Solution
8. Date Re-tested

The test team leader will provide weekly updates of completed scripts, success/correction percentages to project leader, technical and functional team leads. The test team leader will also update the project plan and provide written notice to Development Lead of repeated errors.

6 Testing Schedule

The test schedule section will cover the overall LaRS test schedule, security, milestone information, personnel requirements and deliverable materials.

6.1 Overall Test Schedule

The LAP and LaRS testing workplan in Section 9.1 reflects the time duration of Unit Test, System Test, Integration Test (LaRS only), and User Acceptance Test. The workplan includes tasks, percentage (%) complete, duration, begin and end dates, and resource allocation

6.2 Security

Once the initial System Test environment is created, the LaRS Development Team will assist in the testing effort but not have access to the System Test environment. Only members of the System Test Team will have access to the environment. Migration of system components from the System Test promotion group in the Configuration Management Tool to the System Test environment will be controlled and performed solely by the System Test Team. This will contribute to a valid System Test by ensuring that no one outside the System Test Team has access to the System Test environment. In addition, System Testers will have sole ownership of the System Test environment to ensure that new versions of code are not introduced during passes of test execution.

The testers conducting System, Integration, and User Acceptance Tests will use a set of internal and external user IDs. This will allow the System Test Team to verify the system security procedures and test connectivity.

6.3 Mile Stone Chart

The following table depicts the activities and events to be conducted for Unit, System, Integration, and User Acceptance Tests.

Phase	Date
Test Script Development	4/25 – 6/3
TRR	5/30
System Test	6/3 – 7/26
Integration Test	7/15 – 8/9
UAT	8/12 – 8/23
Production Simulation	8/12 – 8/23

Pre PRR	8/26
PRR	8/28
Submit CR	8/28
LaRS Production Release	9/9

6.4 Personal Requirements

The following table provides a listing of the community personnel necessary to complete the LAP and LaRS User Acceptance testing effort.

Name	**Title**	**Phone #**	**Email**	**Location**	*Time Required*

6.5 Deliverable Materials

The following Unit, System, Integration, and User Acceptance Test deliverables will be delivered to the client for review as part of the LAP and LaRS testing efforts:

- Testing Plan (Delivered prior to the testing effort)
- Test Results and Evaluation Report (Delivered after to the testing effort)

7 Assumptions, Issues, and Action Items

7.1 Assumptions

This document assumes the following for the duration of the testing effort. If any of these assumptions prove to be false, changes will result to the approach, the schedule, or both.

1. The detailed functional design will accurately describe the workings of the system, and the software will adhere to the design.
2. The environment in which functional testing will be performed will be fully available to the testing team for the duration of this effort, as defined in this document.
3. The testing team will have access to a fully documented subsystem prior to writing test scripts. Documentation will include process flows and numbered lists of business rules (by screen and across screens).
4. System response time will be within acceptable tolerance (*e.g.*, transactions should post in a matter of seconds, not minutes).
5. Certain key components of the new AR subsystem, such as the packages that post records to the general ledger and the AR history table, will be among the last developed. Therefore, the programs that call these packages cannot be considered fully tested until these packages are complete. This will involve regression testing of all affected areas.

7.2 Issues

- None at present.

7.3 Action Items

- Iyer will setup the Test Script database.

8 APPENDICES

8.1 Testing Workplan

<Insert>

8.2 Resource Estimate

	Scenario/Script Design & Develop					**System Test Execution**				
	Count		Effort (days)	Duration (days)	FTE's	**Count**		Effort (days)	Duration (days)	FTE's
Interfaces										
High	0	3	0			0	4	0		
Medium	4	2	8			0	3	0		
Low	0	1	0			5	1.5	7.5		
Total	***4***		***8***	***4***	***2.0***	***5***		***7.5***	***4***	***1.9***
Conversions										
High	0	3	0			0	4	0		
Medium	0	2	0			1	3	3		
Low	1	1	1			0	1.5	0		
Total	***1***		***1***	***3***	***0.3***	***1***		***3***	***3***	***1.0***
AR/AP										
High	1	3	3			0	4	0		
Medium	0	2	0			1	3	3		
Low	0	1	0			0	1.5	0		
Total	***1***		***3***	***5***	***0.6***	***1***		***3***	***5***	***0.6***
Extensions										
High	7	3	21			7	4	28		
Medium	15	2	30			26	3	78		
Low	21	1	21			9	1.5	13.5		
Total	***43***		***72***	***20***	***3.6***	***42***		***119.5***	***25***	***4.8***
Reports										
High	0	3	0			0	4	0		
Medium	0	2	0			0	3	0		
Low	6	1	6			6	1.5	9		
Total	***6***		***6***	***3***	***2.0***	***6***		***9***	***3***	***3.0***

					8.2					10.3

LaRS Testing Estimate

			Scenario/Script Design			Sys Test Execution				
			Low	Med	High	Low	Med	High		Resource
Workunits:										
INTERFACE						1				
Peps				1		1				Tester 2
NSLDS				1		1				Tester 2
Bureau of Census Interface				1		1				Tester 2
LaRS Summary Table Population				1		1				Tester 2
Interface Total			0	4	0	5	0	0		
CONVERSION										
LAP to LaRS			1				1			Iyer
Conversion Script Total			1	0	0	0	1	0		
AR/AP										
Summary Table Extract					1		1			Tester 2
Conversion Script Total			0	0	1	0	1	0		
EXTENSIONS										
Admin			L	M	H	L	M	H		
Responsibility/Security				1			1			Iyer
LAP Approval			L	M	M	L	M	H		
Approval				1			1			Tester 2
Create LID			1			1				Tester 2
View Details			1			1				Tester 2
LaRS Lender Screens - 799			L	M	H	L	M	H		
Activities				1			1			Iyer, Tester 1
Loan Origination/Edits					1			1		Iyer, Tester 1
Interest Benefits/Edits					1			1		Iyer, Tester 1
Special Allowance/Edits					1			1		Iyer, Tester 1
Loan Activity/Edits					1			1		Iyer, Tester 1
Loan Portfolio Status/Edits					1			1		Iyer, Tester 1
Disclaimer			1			1				Tester 1
Deactivation Candidates			1			1				Tester 1
Deactivation Candidates - Workflow			1			1				Tester 1
LaRS Lender Screens - Supplier			L	M	H	L	M	H		

Status										
Hold				1			1			Iyer, Tester 1
Deactivation			1			1				Iyer, Tester 1
Supplier LOV Maintenance			1			1				Iyer, Tester 1
Interest Rate Maintenance			1			1				Iyer, Tester 1
Profile			L	M	H	L	M	H		
Contact			1	1			1			Tester 2
Payment				1			1			Tester 2
GA				1			1			Tester 2
Servicer Info				1			1			Tester 2
Servicer Lender				1			1			Tester 2
Entity Info				1			1			Tester 2
Additional LIDS				1			1			Tester 2
LaRS ED Screens			L	M	H	L	M	H		
Table 17				1			1			Iyer
Table 18				1			1			Iyer
Table 30			1			1				Iyer
Notices			L	M	H	L	M	H		
Reasonability			1				1			Tester 2
File Transfer Reject			1				1			Tester 2
LaRS Error Notice - Paper/Email			1				1			Tester 2
Lender Candidate Deactivation			1				1			Tester 2
Lender Deactivation			1				1			Tester 2
LID Activation			1				1			Tester 2
Lender Servicer Profile Change			1				1			Tester 2
File Transfer Confirmation			1				1			Tester 2
LAP Reject			1				1			Tester 2
LAP Accept w/OPA			1				1			Tester 2
LAP Accept w/o OPA			1				1			Tester 2
File Transfer			L	M	H	L	M	H		
Hourly pull from SAIG Mailbox to FMS					1			1		Iyer, Tester 1
File Receipt Notification				1			1			Tester 1
Invoice Level Validation				1			1			Iyer, Tester 1
Processing Notification					1			1		Tester 1
EXTENSIONS GRAND TOTAL			21	15	7	9	26	7		
REPORTS			L	M	H	L	M	H		
GA Association Report			1			1				Tester 2
Servicer Association Report			1			1				Tester 2
Trustee Association Report			1			1				Tester 2
Lender Association Report			1			1				Tester 2
Reasonability Test Summary Report			1			1				Tester 2

	Lender Report Views			1			1				Tester 2
	Lender Search Report			1			1				Tester 2
	Late Filers Report			1			1				Tester 2
	Deactivation Candidate report			1			1				Tester 2
	Deactivation Report			1			1				Tester 2
	Activity Summary Report			1			1				Tester 2
	Quarterly Summary Report			1			1				Tester 2
	Tax Exempt Interest Payments report			1			1				Tester 2
	Edits Analysis Report			1			1				Tester 2
	Lender Analysis Report			1			1				Tester 2
Reports Total				6	0	0	6	0	0		
GRAND TOTAL				27	19	8	20	27	7		

8.3 Sample Test Script

Script Number: LR_Profile_04
Description: Script to test the LaRS Profile Page
Created By: Todd Collins
Created Date: 04/16/2002
Modified Date: TJC 4/16/02
Version: 1.2
Tested By:
Test Date:
Prerequisites:
Conditions Covered:

Step	Cond. Num.	Test Condition	Action	Expected Results	Actual Results	Pass/ Fail	Comments	SIR #
1.	1.1.1	Profile screen can be entered from the Navigator Menu.	From the Navigator Menu, double-click on the Profile screen.	The Profile detail screen is displayed.				
2.	1.2.1	First Name Can Be Changed	Enter the Following information: Bill	System should have the following default values: First Name should be changed				

Ex

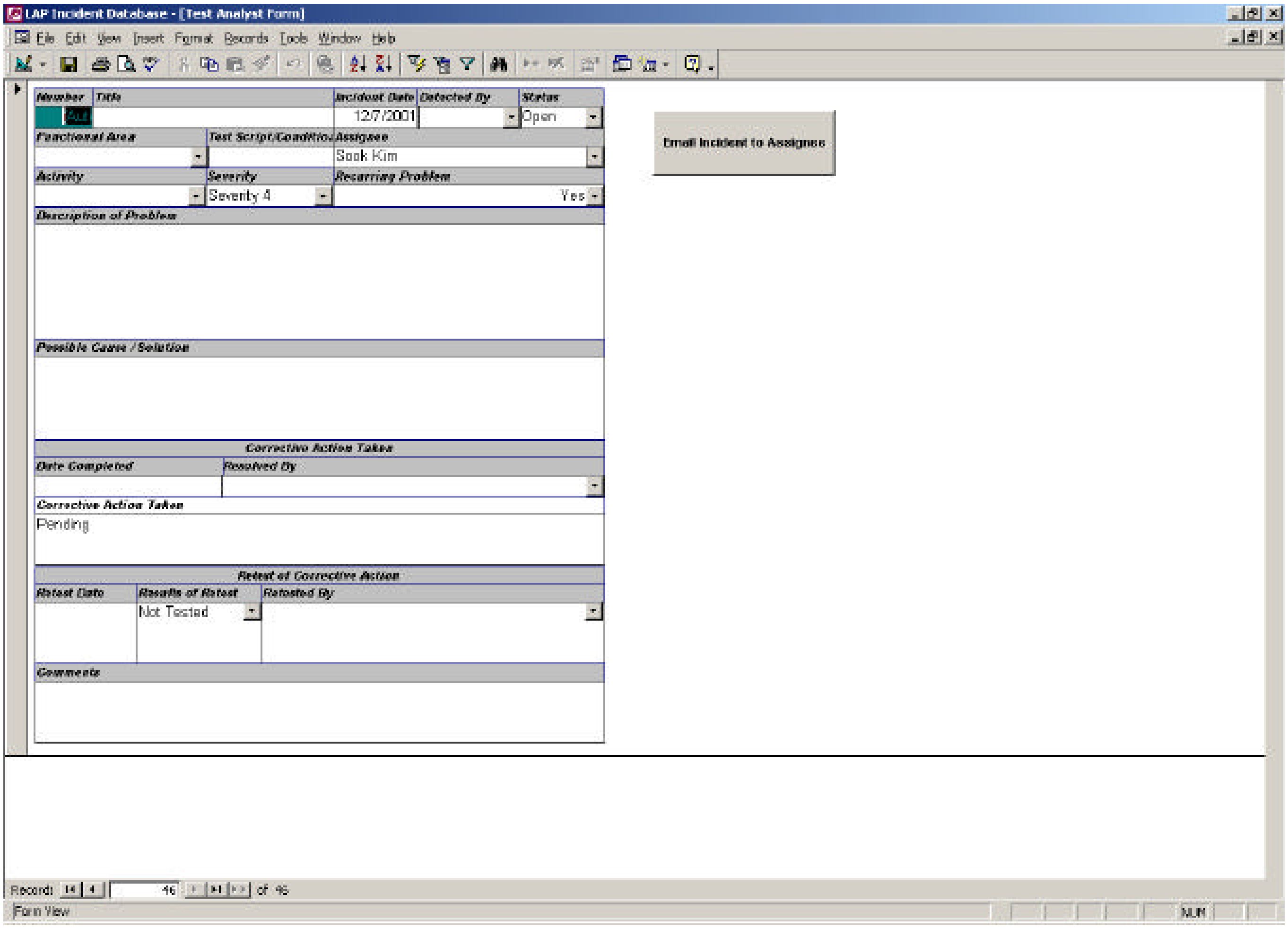
1. The Test Condition Column contains the actions a tester should perform to execute the test condition.

2. The Expected Results Column contains signs that a tester can use to verify that the test was successful. This could be a visual cue such as “field is highlighted”, “error is returned”, “welcome message is displayed”, etc.

3. The Test Result (Pass/Fail) Column is used by the tester to document the actual results during test execution.

4. When the actual results do not coincide with the expected results, the Comments Column is used by the tester to document an explanation detailed enough so it can be recreated and fixed.

8.4 Incident Database – Test Analyst Form

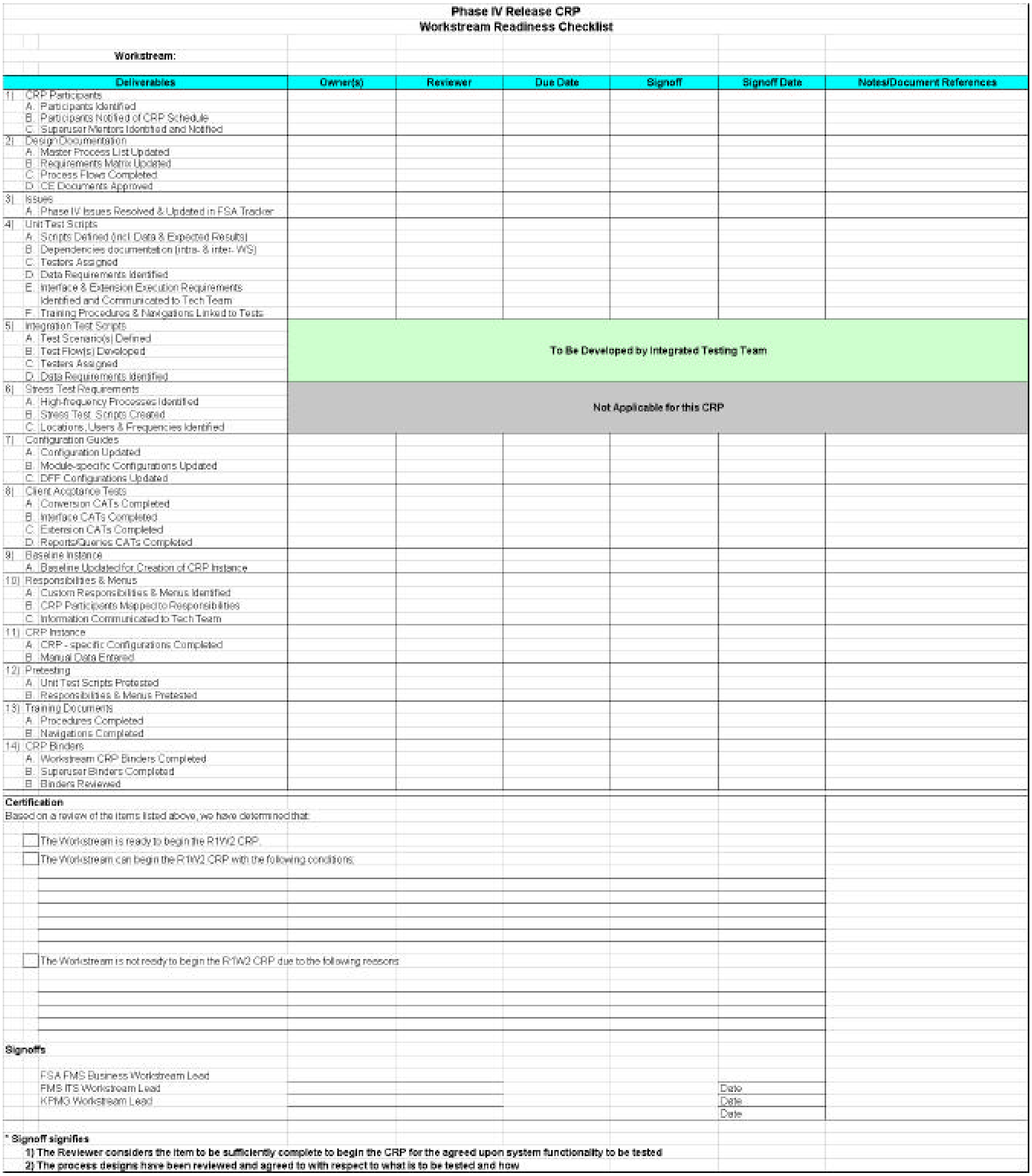

LAP Incident Database - [Test Analyst Form]
File Edit View Insert Format Records Tools Window Help
Email Incident to Assignee
Record: 46 of 46
Form View
NUM

Number: Aut
Title:
Incident Date: 12/7/2001
Detected By:
Status: Open
Functional Area:
Test Script/Condition:
Assignee: Sook Kim
Activity:
Severity: Severity 4
Recurring Problem: Yes
Description of Problem:
Possible Cause / Solution:

Corrective Action Taken
Date Completed:
Resolved By:
Corrective Action Taken: Pending

Retest of Corrective Action
Retest Date:
Results of Retest: Not Tested
Retested By:
Comments:

8.5 Workstream Readiness Checklist



Phase IV Release CRP
Workstream Readiness Checklist

Workstream:

Deliverables	**Owner(s)**	**Reviewer**	**Due Date**	**Signoff**	**Signoff Date**	**Notes/Document References**
1) CRP Participants						
A. Participants Identified						
B. Participants Notified of CRP Schedule						
C. Superuser Mentors Identified and Notified						
2) Design Documentation						
A. Master Process List Updated						
B. Requirements Matrix Updated						
C. Process Flows Completed						
D. CE Documents Approved						
3) Issues						
A. Phase IV Issues Resolved & Updated in FSA Tracker						
4) Unit Test Scripts						
A. Scripts Defined (incl Data & Expected Results)						
B. Dependencies documentation (intra- & inter- WS)						
C. Testers Assigned						
D. Data Requirements Identified						
E. Interface & Extension Execution Requirements Identified and Communicated to Tech Team						
F. Training Procedures & Navigations Linked to Tests						
5) Integration Test Scripts	**To Be Developed by Integrated Testing Team**					
A. Test Scenario(s) Defined						
B. Test Flow(s) Developed						
C. Testers Assigned						
D. Data Requirements Identified						
6) Stress Test Requirements	**Not Applicable for this CRP**					
A. High-frequency Processes Identified						
B. Stress Test Scripts Created						
C. Locations, Users & Frequencies Identified						
7) Configuration Guides						
A. Configuration Updated						
B. Module-specific Configurations Updated						
C. DFF Configurations Updated						
8) Client Acceptance Tests						
A. Conversion CATs Completed						
B. Interface CATs Completed						
C. Extension CATs Completed						
D. Reports/Queries CATs Completed						
9) Baseline Instance						
A. Baseline Updated for Creation of CRP Instance						
10) Responsibilities & Menus						
A. Custom Responsibilities & Menus Identified						
B. CRP Participants Mapped to Responsibilities						
C. Information Communicated to Tech Team						
11) CRP Instance						
A. CRP - specific Configurations Completed						
B. Manual Data Entered						
12) Pretesting						
A. Unit Test Scripts Pretested						
B. Responsibilities & Menus Pretested						
13) Training Documents						
A. Procedures Completed						
B. Navigations Completed						
14) CRP Binders						
A. Workstream CRP Binders Completed						
B. Superuser Binders Completed						
B. Binders Reviewed						

Certification

Based on a review of the items listed above, we have determined that:

□ The Workstream is ready to begin the R1W2 CRP.

□ The Workstream can begin the R1W2 CRP with the following conditions:

□ The Workstream is not ready to begin the R1W2 CRP due to the following reasons:

Signoffs

FSA FMS Business Workstream Lead
FMS ITS Workstream Lead
KPMG Workstream Lead

Date
Date
Date

*** Signoff signifies**

1) The Reviewer considers the item to be sufficiently complete to begin the CRP for the agreed upon system functionality to be tested

2) The process designs have been reviewed and agreed to with respect to what is to be tested and how